UNITED STATES LETTERS PATENT APPLICATION
FOR:

METHOD AND SYSTEM FOR REPRESENTING, CONFIGURING AND DEPLOYING DISTRIBUTED APPLICATIONS

INVENTORS:

Markus Schmidt-Karaca
Janusz Smilek
Sven Bernhard

Prepared by:

Kenyon and Kenyon

KENYON & KENYON
1 Broadway
New York, New York 10024
(212) 425-7200

METHOD AND SYSTEM FOR REPRESENTING, CONFIGURING AND DEPLOYING DISTRIBUTED APPLICATIONS

FIELD OF THE INVENTION

The present invention relates to the fields of information, computer software systems and computer networks. In particular, the present invention provides a method and system for representing, configuring and deploying distributed applications in any arbitrary network topology.

BACKGROUND INFORMATION

The complexities of modern business practices have required an evolution in computing networks and associated software systems. For example, distributed software systems in which a software application solution is achieved via a collaborative effort between multiple computing entities within the network have become commonplace.

As depicted in FIG. 1a, distributed software systems present particular administrative, configuration and deployment challenges. For example, a particular application solution may involve the roles of many computing entities having heterogeneous capabilities 181b. In particular, the complexity of modern computer networks is complicated by the heterogeneous nature of devices connecting to the network. For example, modern networks must allow simultaneous connection of desktop computers, laptop computers, PDAs ("Personal Digital Assistants") etc. Further, each of these device classes may utilize different operating systems, and may interface with peripheral devices in any arbitrary manner. Each computing entity participating in an application solution may require a combination of many resources including executable program code, data, etc. In complex networks, thousands of distributed applications may coexist.

Second, the resources and applications comprising the distributed applications may also be evolving in time as software developers generate updates and patches (181a). Thus, deployment of resources within a distributed computing environment is complicated due to concurrent software development.

Third, network users themselves also impose a diverse set of requirements 181c. For example, some users may require particular configuration of their devices such as particular power saving schemes, enablement of disablement of various menu items or options, particular options for installed applications, etc.

Managing such a diverse set of devices among an equally diverse user-base becomes a logistically complex. The complexity of the problem is further complicated by the simultaneous development efforts, through which updates, patches, are evolving in real time.

SUMMARY OF THE INVENTION

The present invention provides a method and system for representing, configuring and deploying a distributed application solution in any arbitrarily complex network. According to one embodiment of the present invention, for each distributed application solution, an application solution descriptor ("ASD") maintains a representation of all resources associated with computing entities participating in the application solution. The ASD may be utilized to perform automatic configuration of the participating computing entities as well as deployment of resources to particular computing entities.

According to one embodiment the present invention is applied to a mobile computing environment including a backend server, a middleware server and any number of mobile devices. Each mobile application solution includes collaborative behavior between a mobile device, the middleware server and the backend server. Accordingly, for a particular mobile application solution, a mobile device, middleware server and backend server are associated with respective resources that need to be configured and/or installed in order for the application to run.

According to one embodiment of the present invention, a MSD ("Mobile Solution Descriptor") maintains a representation of all mobile application solutions and the respective resources required for those applications with respect to the backend server, middleware server and mobile device. According to one embodiment, the MSD is stored in a central location on the middleware server and is utilized to provide automatic configuration of the mobile device, the middleware

server and the backend server as well as deployment of resources to computing entities.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1a depicts various deployment and configuration challenges in a network environment.

FIG. 1b depicts an abstract representation of a computing environment with respect to a number of exemplary application solutions according to one embodiment of the present invention.

FIG. 2 depicts an embodiment of the present invention as applied to a mobile environment according to one embodiment of the present invention.

FIG. 3 shows a class diagram for an application solution descriptor according to one embodiment of the present invention.

FIG. 4 depicts a configuration and deployment process according to one embodiment of the present invention.

FIG. 5 illustrates the operation of a configuration and deployment module according to one embodiment of the present invention.

FIG. 6 is a flowchart illustrating the use of a mobile solution descriptor for configuration of one or more application solutions.

DETAILED DESCRIPTION

FIG. 1b depicts an abstract representation of a computing environment with respect to a number of exemplary application solutions according to one embodiment of the present invention. Each of computing entities 101(1)(1)-101(J)(N) provides a locus for computational activity or processing tasks having and has an associated set of hardware dependent capabilities. Computing devices 101(1)(1)-101(J)(N) may be

servers, desktop computers, laptop computers, personal desktop assistants, etc. Each computing device includes one or more processors and may be associated with one or more storage devices (not shown) such as a hard drive, CD ROM drive, etc. Computing devices 101(1)(1)-101(J)(N) collectively represent a heterogeneous set of computing resources.

As shown in FIG. 1b, computing devices 101(1)-101(N) may be situated in any arbitrarily complex network topology. Thus, computing devices 101(1)-101(N) may be co-located, disparately located or any combination of the two and may be situated on any number of disparate networks 130(1)-130(J) utilizing any number of network protocols (not shown). For example, as shown in FIG. 1, computing devices 101(1)(1)-101(1)(K) are situated on network 130(1), computing devices 101(2)(1)-102(2)(L) are situated on network 130(2), computing devices 101(3)(1)-101(3)(M) are situated on network 130(3) and computing devices 101(J)(1)-101(J)(N) are situated on network 130(J).

FIG. 1b also shows a number of exemplary application solutions 150(1)-150(4). Each application solution 150(1)-150(4) includes a subset of computing devices 101(1)(1)-101(J)(N) performing a defined collaborative behavior. Thus, for example, application solution 150(1) includes the collective behavior of computing devices 101(3)(M), 101(1)(K) and 101(2)(L). Application solution 150(2) includes the collective behavior of computing devices 101(3)(1), 101(K) and 101(2)(L). Application solution 150(3) includes the collective behavior of computing devices 101(J)(N) and 101(2)(2). Application solution 150(4) includes the collective behavior of computing devices 101(2)(2), 101(J)(N), 101(J)(2) and 101(J)(1).

Note that the same computing device 101 may participate in multiple application solutions 150. For example, computing device 101(1)(K) participates in application solutions 150(1) and 150(2).

In order to define the particular behavior of a computing device 150 with respect to a particular application solution 150, at least one application solution resource is installed on that device. For example, with respect to application solution 150(1), application solution resource 140(2)(1) is installed on computing device 101(3)(1). On the other hand, since computing device 101(2)(L) participates in both application solutions 150(1) and 150(2), application solution resource 140(1)(4), which is associated with application solution 150(1) and application solution resource

140(2)(3), which is associated with application solution 150(2) are both installed on computing device 101(2)(L).

Effectively, a particular application solution 150 defines a distributed application, wherein intermediate computation data, instructions, etc. may be transmitted between and among participating computing devices 101. It is understood, that communication between computing devices 101 may be achieved using any type of network protocols and/or network protocol negotiation if particular computing devices reside on different types of networks.

FIG. 2 depicts an embodiment of the present invention as applied to a mobile environment according to one embodiment of the present invention. Processing tasks comprising a mobile application solution are performed on computing entities backend server 301, middleware server 302 and mobile devices 315(1)-315(N). Mobile devices 215(1)-215(N) may include laptop devices, PDAs or any type of device that may be desirable for use within a mobile environment.

FIG. 2 also shows an arbitrary number of mobile application solutions 350(1)-350(N). Each application solution 350(1)-350(N) represents the collective behavior of backend server 301, middleware server 302 and one respective mobile device 315(1)-315(N). Thus, for example, the collective behavior of mobile device 315(2) (in this case a PDA), middleware server 302 and backend server 301 comprises mobile application solution 350(2). Similarly, the collective behavior of backend server 301, middleware server 302 and mobile device 315(3) (in this case a laptop computer) comprises mobile application solution 350(3).

In general, mobile devices 315(1)-315(N) have limited processing and memory capabilities and thus require the services of backend server 301 and/or middleware server 302 to carry out a mobile application solution. Each mobile device 315(1)-315(N) may include a local database (not shown). Thus, as shown in FIG. 2, each mobile application solution 350(1)-350(N) includes the computing entities of a particular mobile device (e.g., 315(2)) as well as backend server 301 and middleware server 302. Backend server operates as an application server and thus provides services of processes 301(1)-301(N) for data processing. Data obtained at mobile devices 315(1)-315(N) may be transferred to backend server 301 via middleware server 302 for processing and subsequently may be returned to mobile devices 315(1)-315(N) post-processing. Thus, as shown in FIG. 2, backend server 301 executes a

plurality of processes 303(1)-303(N). Each of these particular processes may participate in a particular mobile application solution 350(1)-350(N) or not depending upon the particular application.

Middleware server 302 provides functions of routing via routing module 445 processing requests received from particular mobile devices 315(1)-315(N) to backend server for processing via particular processes 303(1)-303(N). In addition, middleware server performs routing of processed data from backend server 301 to particular mobile devices 315(1)-315(N) with respect to a particular mobile application solution (i.e., 350(1)-350(N)). In addition, middleware server 302 performs replication of data for a particular mobile application solution (350(1)-350(N)), which is received from backend server 301. Thus, as shown in FIG. 2, middleware server 302 receives and stores respective replicated data 325(1)-325(N) from backend server. This data is then employed as part of a particular mobile application solution 350(1)-350(N).

In addition, middleware server 302 performs a number of administrative tasks including configuration and deployment via configuration module 430, deployment module 440 and mobile solution descriptor 475. Mobile solution descriptor 475 stores a complete representation of all mobile application solutions 350(1)-350(N) existing on the network.

Mobile devices 315(1)-315(N) may operate in either an online or offline mode. However mobile devices 315(1)-315(N) provide a transparent experience to the user as if the devices were online at all times. This is achieved by providing resources 140(1)(1)-140(1)(K), 140(2)(1)-140(2)(L), 140(3)(1)-140(3)(M) and 140(N)(1)-140(N)(J) to respective mobile devices 315(1)-315(N). These resources provide processing and user interface behavior on the mobile device to so that the user experience is as if the respective application solution 350(1)-350(N) were operating locally on the respective mobile device. If a particular mobile device (315(1)-315(N)) is operating in an offline mode, that mobile device caches data locally on a local database (not shown), which can later be synchronized with backend server 302.

Each mobile device 315(1)-315(N) is further equipped with a respective configuration module 333(1)-333(N), which functions in conjunction with configuration module 430 on middleware sever 302 during a configuration process (described below). The purpose and function of respective configuration modules

333(1)-333(N) on mobile devices 315(1)-315(N) will become evident as the invention is further described.

Middleware server 302 includes routing module 445, configuration module 430, deployment module 440 and mobile solution descriptor 475. Configuration module 430 functions in conjunction with respective configuration modules 333(1)-333(N) on mobile devices 315(1)-315(N) to perform configuration processes with respective devices. Mobile solution descriptor 475 maintains a representation of all mobile application solutions (i.e., 350(1)-350(N)) existing within the network. In particular, as described below, mobile solution descriptor 475 maintains a representation of all resources, configuration settings relating to each mobile application solution 350(1)-350(N). In particular, with respect to the present invention, mobile solution descriptor 475 maintains a representation of configuration settings such as menu options, power settings, required resources, etc. for each of respective mobile devices 315(1)-315(N) participating in a particular mobile application solution 350(1)-350(N). Thus, for example, mobile solution descriptor 475 may maintain a representation of a registry for each mobile device 315(1)-315(N) – in particular, the resources that are current installed on the mobile device with respect to the associated mobile application solution.

Middleware server 302 also includes configuration module 430 and deployment module 440. Configuration module 430 performs configuration processes as a function of mobile solution descriptor 475 in conjunction with configuration modules 333(1)-333(N) on respective mobile devices 315(1)-315(N). In particular, as described below, mobile solution descriptor 475 maintains a representation of all resources necessary for a particular mobile application solution and thus the resources that must be installed on a particular mobile device. Similarly, deployment module 440 performs deployment processes to transfer resources and data to particular mobile devices 315(1)-315(N).

FIG. 3 shows a class diagram for an application solution descriptor according to one embodiment of the present invention. The class architecture shown in FIG. 3 may be applied to a general computing environment as shown in FIG. 1b or the particular mobile environment (to represent a mobile application solution) as shown in FIG. 2. Referring to FIG. 3, application solution descriptor 475 includes application solution class 377, computing entity class 379 and resource class 393.

Application solution class 377 is associated with a unique identifier 385 for identifying a particular application solution. Application solution class 377 is associated with at least one computing entity via computing entity 379 and at least one resource class 393. Each instance of computing entity class 379 includes a unique identifier 390 as does each instance of resource class 395. Each instance of a resource class 393 is associated with a particular computing entity ID 390, which indicates a particular computing entity (e.g., 315(1)-315(N)) upon which the resource should be installed.

FIG. 4 depicts a configuration and deployment process according to one embodiment of the present invention. FIG. 4 depicts a deployment process with respect to only two mobile devices 315(1) and 315(2). However, it is to be understood that the process may be applied to any arbitrary number of mobile devices. Referring to FIG. 4, backend server 301 stores application data 480 and application resources 490. Application data 480 and application resources are ultimately deployed to mobile devices 315(1) and 315(2) via middleware server 302. The configuration and deployment process is accomplished via configuration module 430 and deployment module 440 on middleware server and respective configuration modules 333(1) and 333(2) on respective mobile devices 315(1) and 315(2).

FIG. 5 illustrates the operation of a configuration and deployment module according to one embodiment of the present invention. FIG. 5 depicts configuration and deployment with respect to a single computing entity(315(1)). However, it is to be understood that the invention may be applied to any number of computing entities as shown in the preceding figures. Configuration module 430 on middleware server 302 communicates with communication module 333(1) on computing entity 333(1). During a synchronization process, computing entity 315(1) sends synchronization request to configuration module 430 on middleware server 302. Synchronization request 550 includes various information including the ID of computing entity 315(1). Upon receipt of synchronization request 550, configuration module attempts to determine appropriate resources to be deployed to computing device 315(1). In order to determine the appropriate resources, configuration module queries application solution database 435 using computing entity ID 390. Application solution database 435 returns resource ID 510 as a function of received computing entity ID 390.

In order to generate a particular resource ID as a function of computing entity ID 390, application solution database 435 matches computing entity ID 390 to a particular mobile application solution 350 using the class structure shown in FIG. 3. In particular, this is achieved using mobile solution descriptor 475. Mobile solution descriptor maps a particular computing entity ID 390 to a particular resource type ID 395. In addition, configuration parameters for the mobile application solution 350 are retrieved from configuration parameters table 477. Finally, the device profile associated with computing entity ID 390 is determined. A particular resource ID 510 is then generated as a function of the retrieved device profile, application solution and associated configuration parameters for that device and returned to configuration module.

Configuration module 510 then forwards the particular resource ID 510 to deployment module 440. Deployment module utilizes the received resource ID 510 to retrieve the appropriate application resource for the particular computing entity 315(1) from application resources database 490. The resource (e.g., 140(1)(1) . . . 140(1)(K)) are then forwarded to computing entity via configuration module 333(1). Configuration module 333(1) on computing entity 315(1) receives the resources and installs them on computing entity 315(1).

Computing entity 315(1) also includes registry 450(1). During the configuration process as outlined above, the registry information may be transmitted to configuration module 430 from configuration module 333(1) to determine whether particular resources (e.g., 140(1)(1)-140(1)(K)) need to be updated

FIG. 6 is a flowchart illustrating the use of a mobile solution descriptor for configuration of one or more application solutions. The process is initiated in step 610. In step 620, the MSD is uploaded from a database (not shown). According to one embodiment, the MSD is stored on middleware server. The MSD is then utilized to determine which resources need to be installed and configured on each mobile device, the middleware server and the backend server for each application solution. In step 625, communications are initiated with backend server. In step 630, data necessary for the application is replicated from the backend server to the middleware server. In step 635, resources necessary for the application are downloaded from the backend server to the middleware server. In step 640, the middleware server is configured to perform various tasks related to the application solution such as the

forwarding of data received from a mobile device to the appropriate application process running on the backend server. In step 645, it is determined whether a mobile device has connected to the middleware server. If so, in step 650, necessary data and resources for execution of the application solution are downloaded to the connecting mobile device. In step 655, the configuration is complete and the process ends.